Week 2 - Part 1 : Engineering Data Science Systems

☰ pending tasks	
☰ type	

Engineering aspects of DS

2 types of contents:

a. Math behind DS (Algorithmic section)

b. Engineering aspects of it (building a system that can be deployed), includes conceptual ideas for software engineering(SE) and programming.

What is systems thinking?

It is a way of thinking of systems that is global and encompassing, rather than focused on particular issues. Knowing the larger context of the application module being developed is systems thinking.

System perspective of DS

DS has a very broad context and thus, systems thinking is quintessential.

Roles that come together for DS to work :

1. Domain knowledge : understanding the context that generates the data.
2. Mathematics and statistics
3. Hacking skills : engineering and programming skills

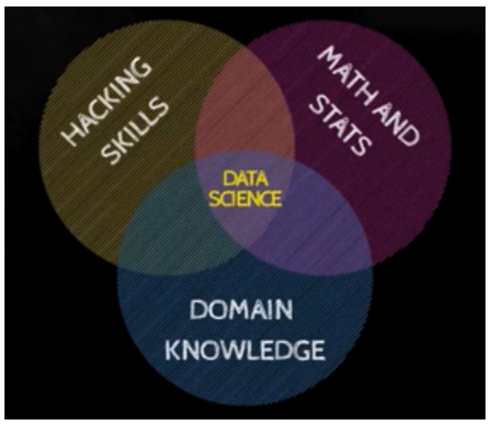

Fig.1 What forms DS.

Workplace skills required:

- Business knowledge (domain knowledge)
- Programming skills
- Knowledge of Statistics
- communication skills : using visualization techniques

Job profiles and respective Skills :

1. Data analyst : business understanding + statistical programming + communication skills, important in initial stages of DS projects. a.k.a DS consultants.
2. Research Engineer : programming + statistics + communication skills, might not know the nuances of business aspects. These are the people who accelerate innovation.
3. Data Engineer : business understanding + programming skills, manage the IT infrastructure required for DS projects.

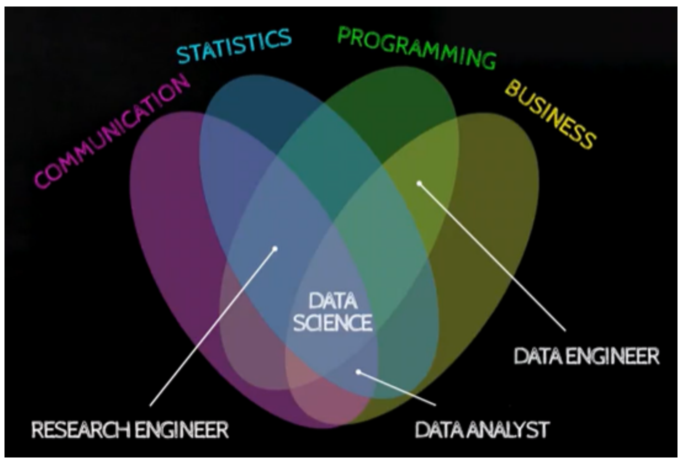

Fig.2 job profile and skill set required.

CRISP-DM_Business Understanding

Engineering systems for DS involves the two components of **process and programming.** Process is a flow of steps and uses the idea of agile improvement. Agile improvement gives out a working product at the end of each iteration. This product is incrementally improved.

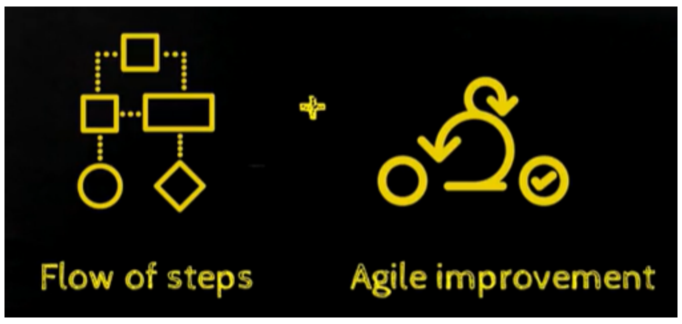

Fig. 3 steps in process.

Example process followed for DS : CRISP-DM (seeing what steps involved and improving them in an agile way) stands for Cross-Industry Standard Process for Data Mining.

Steps of DS:

DS requires data.

1. Business understanding : understanding the business objectives and problems.
2. Data understanding : checking if the data in hand can be used to meet the business objectives. Also, collecting new data.
3. Data preparation
4. Data modeling
5. Evaluation with multiple data
6. Deployment or re-iterate the process.

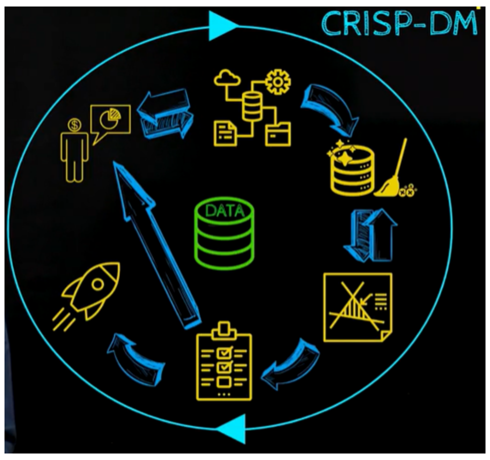

Fig.4 CRISP-DM model of development.

STEP 1 : Business Understanding

The following five questions are to be answered to get a better understanding of business. This is done iteratively and refined, thus, bringing in agile development within business understanding.

1. **What are the business objectives?**

Capture the dimensions to be optimized and the resources available in a systematics way.

2. Can DS achieve the objectives?

Not all objectives require the DS to solve the problem.

3. How is success metric defined?

For robust understanding of the output required, defining a success metric is important.

4. What are ethical considerations in data usage?

Checking if there are violations of privacy and legal considerations whilst using the data.

5. What have other industries achieved?

Finding out what the State of the Art (SOTA) is.

CRISP-DM_Data Understanding, Preparation and Modelling

STEP 2: Data Understanding

1. What are the sources of data?
2. Does new data need to be collected?
3. What is the quantity and quality of data available?
4. What do different data items represent?
5. Which data is relevant to the objective?

STEP 3 : Data Preparation

The different stages of data preparation revolves around answering the following questions:

1. **What are the different data formats?**

How to bring data from different sources to a standard format

2. Is there a need to annotate data?

This is addition of labels by humans. Manual task.

3. How can data be extracted, transformed and loaded?

Also known as ETL. Requires programming skills.

4. How to standardise and normalise data?

Standardise : 0 mean and unit variance

Normalise : values lie between 0-1

5. How to efficiently store data for analysis?

STEP 4 : Data Modelling

The following five agile steps are to be used in modeling the data :

- **What are the assumptions made about the data?**

The assumptions made about the data, based on the domain knowledge are to be specified.

- **Statistical or algorithmic modelling ?**

Decide which approach suits the business context.

- **Is clean data sufficient for modelling?**

If not, go back to the step of data preparation.

- **Is the compute budget sufficient for modelling?**
- **Are results statistically significant?**

This helps to back the results of the model.

CRISP-DM_Evaluation & Deployment

After the model has been developed and tested if it is statistically significant, it is passed on for evaluation before deployment.

STEP 5 : Model Evaluation

- **Does the model work correctly on test data?**

The model is not trained on the test dataset. Thus, is used for evaluation of the results.

- **Does the model achieve business objectives?**
- **Does the model meet performance requirements?**

Checking if the model meets the requirements of the deployment platform.

- **Is the model unbiased and robust?**

The unintentional biases in the model have to be specifically tested and resolved. Robustness of the model against slight changes in the dataset.

- **What are the ways to improve the model?**

STEP 6 : Deployment

1. Where is the model to be deployed?
2. What is the HW/SW stack for deployment?
3. Does it meet performance requirements?
4. Does it violate the privacy requirements?
5. Does it meet users' expectations?

Get the users' feedback.

In each iteration of CRISP-DM, the following take place at a broader level:

1. MVP approach - **Minimal Viable Product** is built at each iteration of product development. It involves iterative design and deployment.
2. Revise expectation of success and value of DS.
3. Upgrade human and hardware resources.

Having a systematic approach to solving problems is much better than ad hoc solution approach.

Programming Tools

- No code environment

H2O.ai, IBM Watson, Amazon Lex, DataRobot

- Spreadsheets, BI tools

Microsoft Excel, Power BI, Google Sheets, Tableau

- Programming languages

Weka, SAS, R, Python, MATLAB, Mathematica

- High performance stack

Hadoop, Spark

Why Python?

- Python is good for prototyping and production. It can easily be understood and written for production.

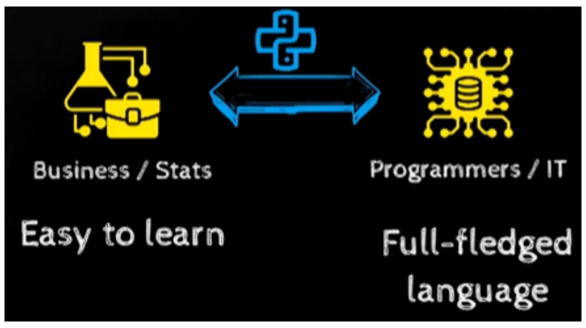

Fig.5 Python for Prototyping and Production (PPP).

- Python is beginner friendly. It is known to be an executable pseudo code.
- Python is increasingly the default choice for DS. There are many open source libraries available in python and maintained well.
- Python can be used as a scripting language to automate the boring tasks. Django can be used for web development. Many IoT devices provide a python interface for programming.

Disadvantage of using python is that it is an interpreted language and not a compiled language. This makes python highly portable but also relatively slower than the compiled counterparts.

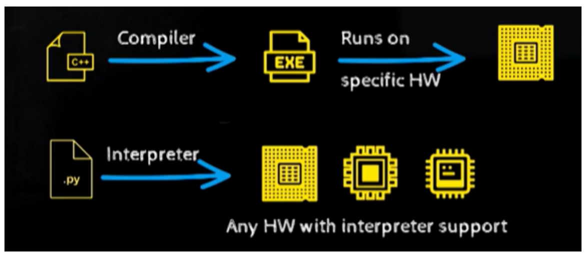

Fig.6 Compiled vs Interpreted language.

Python - Libraries

IPython and Jupyter

IPython encourages iterative read-eval-print (REPL) loop. Jupyter is well suited for documenting the project.

Numpy - Numerical Python (np)

It provides efficient loading, storing and processing of high dimensional data. Numpy arrays can also be used as interfaces with low-level implementation. Inspired from MATLAB.

Pandas (pd)

Used to manipulate structured data. The common objects are dataframe and series. In the five stage pipeline of DS, pandas is very useful in storing, processing and describing data. Inspired from R.

Matplotlib and Seaborn

Used for visualization in python. Works well with Jupyter. Inspired from tools like Tableau.

Scipy

Enables scientific computing with python. Useful in describing data and modelling (scipy.stat). Inspired from MATLAB.

Scikit-Learn

Various ML algorithms are implemented in this library.

Summary

- System thinking helps in systematic development of DS projects.
- There are different job roles within DS that require a combination of business, programming, statistics and communication skills.
- CRISP-DM is a process of DS that aids system thinking.
- There are various programming tools for DS, bucketed by what set of skills are necessary and tasks they are useful for.
- Python is an easy to learn, portable and deployable programming language.

- Pandas, numpy, scikit-learn, scipy, matplotlib and seaborn are some of the most commonly used python libraries for DS.

MCQ : Week2 - Part1

1. In demand forecasting a model predicts the upcoming demand for a product/service. This cannot be used when
    1. **The characteristics of data used to train the model is different from that used to predict.**
    2. **The scenario of prediction is different**
    3. **When the model has a 100% accuracy.**
    4. When the model performance has been tested and validated for two seasons.

2. In CRISP-DM (diagram)
    1. The movement from one phase to the next is unidirectional.
    2. **The back and forth movement between different phases is possible.**
    3. **The arrows in the diagram shows most frequent phase dependencies.**
    4. **The outer circle symbolises cyclic nature of the process**

3. Testing a code in python can be difficult because
    1. It is an interpreted language.
    2. **The variable types are not defined.**
    3. **The return types of the function are not defined and have to be handled correctly.**
    4. All of the above.

4. The libraries used to manipulate the data are
    1. Dataframe
    2. **Pandas**
    3. **Numpy**
    4. matplotlib

5. A systems perspective of DS problem helps in
    1. Understanding the broader context of the modules being implemented.
    2. Understanding the perspectives of different stakeholders.
    3. Facilitating integration of different code units.
    4. **All of the above.**